R & D STATUS REPORT

ARPA ORDER NO:	A418
PROGRAM CODE NO:	DO-C9
CONTRACTOR:	David Sarnoff Research Center
CONTRACT NO.:	N00014-93-C-0202
CONTRACT AMOUNT:	$676,870
EFFECTIVE DATE OF CONTRACT:	18 August 1993
EXPIRATION DATE OF CONTRACT:	17 August 1996
PRINCIPAL INVESTIGATOR:	John C. Pearson (609-734-2385)
TECHNICAL CONTRIBUTORS:	John Pearson, Paul Sajda and Clay Spence
SHORT TITLE:	Hybrid Pyramid / Neural Network Vision System
REPORTING PERIOD:	3/1/95 to 5/31/95

1 The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advance Research Projects Agency of the U.S. Government.

~~2 Ownership in Patent Data included herein is retained by the contractor/subcontractor pursuant to FAR 52.227-12.~~

Description of Progress:

Another Toy Problem for Learning Pattern Trees

An artificial problem was constructed and described in the last report. During this quarter, another, more difficult, artificial problem was constructed. As before, the objects to be found and some potential false positives each have component patterns. Each positive has two different component patterns chosen from three types. In addition, the rectangles which form the objects and the sub-patterns all have artificial shadows added (Figure 1). The angle of the shadow is randomly-chosen. The potential false positive objects either have two component patterns of the same type, or only one pattern of some type, and one or both of the component patterns may not have a shadow. Thus the pattern tree must detect all three types of components with shadows to be able to detect a positive. As in the previous simpler problem, the positives and potential false-positives are 18-by-11 pixel rectangles, but now their brightness is randomly-chosen between 136 to 247. The sub-patterns have a brightness that is independently chosen at random from the same range, but both patterns (if there are two in an object) have the same brightness. The component patterns are three-by-three x, +, and square patterns.

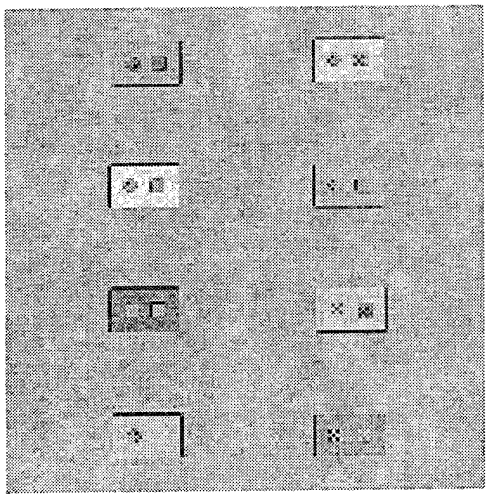

FIGURE 1. Examples of Positives and Potential False Positives. Each positive has two different patterns which are either an "x", a "+", or a square. Positive objects always have shadows. Negatives (lower half) always have one kind of pattern, and may have zero or two patterns, and the patterns do not always have shadows.

A background of noise is added to make identification from residual features at low resolution difficult. The background is Gaussian noise with a spectrum that is approximately proportional to $1/f$. This spectrum is obtained by adding Gaussian white noise to levels 0-4 of a Laplacian pyramid and to level 5, which is treated as an image in the Gaussian pyramid. A full-resolution noise image is constructed from these in the usual way. This gives significant noise at the pyramid levels which will be used in the pattern tree, and significant variation in the background brightness

local to each object. The objects are sometimes brighter and sometimes darker than the surrounding background.

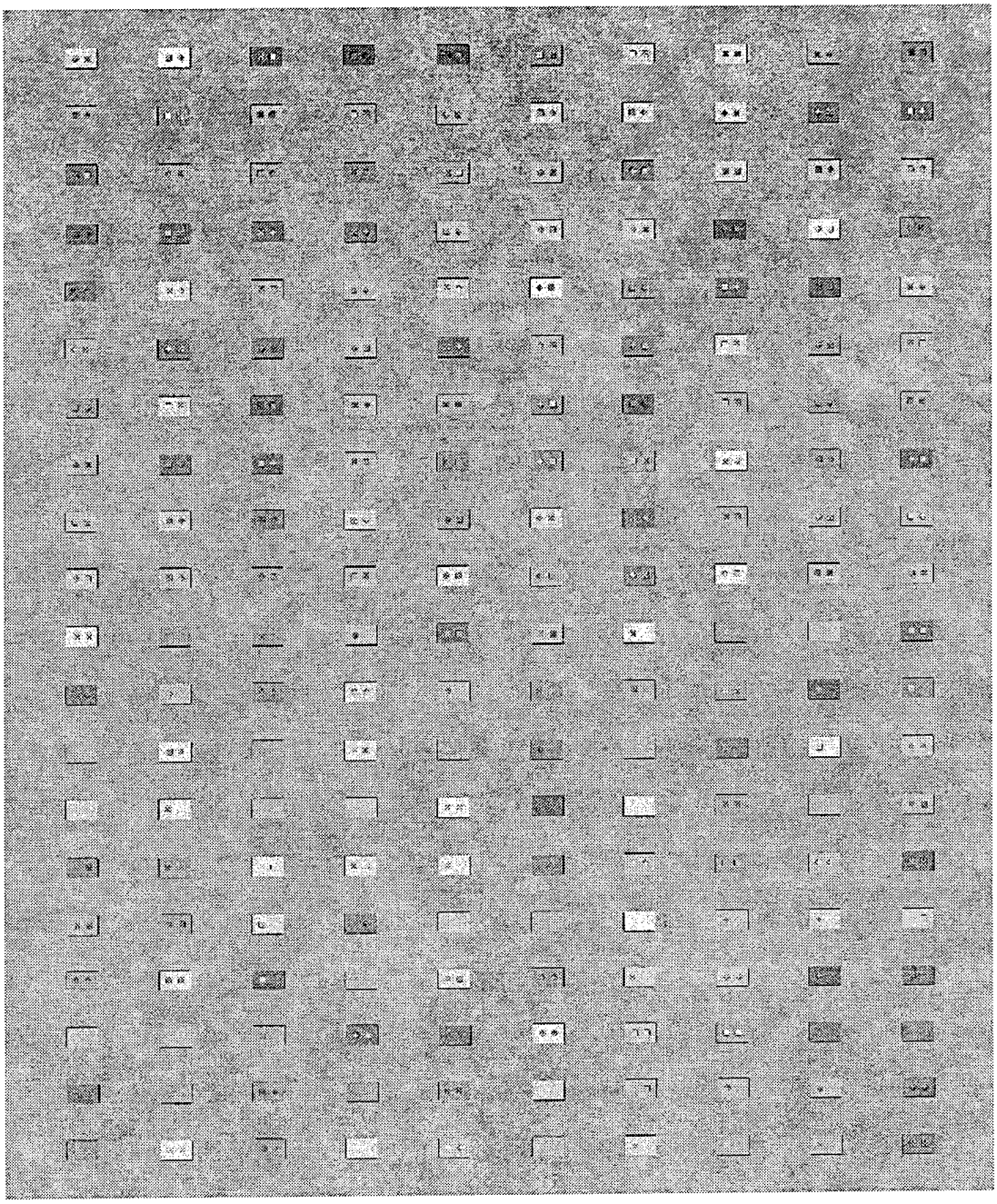

FIGURE 2. Entire Training Image.

The objects, positive and potential false-positive, are arranged in a ten-by-twenty array, with one-hundred positives in the upper half of the image and one hundred negatives in the lower half. The horizontal and vertical spacings between the objects are twice their width and height, respectively, and they are spaced this far from the borders of the image, as well (Figure 2).

A three-level Gaussian pyramid of the training image was made. Each neural net received as input a five-by-five window of pixels from the appropriate pyramid level.

As before, we call those networks which were trained with inputs from the image "component" networks, and those networks with inputs from the component networks "integration" networks. We further refer to them by the pyramid level at which each is used to search, level 0 being the full-size image and level 2 having one-quarter of the linear extent of the full-size image.

For the training of the component networks at levels 2 and 1, the training objective function is a sum over the positive "blobs" of the minimum of the cross-entropy errors at the positions in the blob, plus a sum over negative points of the cross-entropy error at those points. A weight-decay term was added, and roughly optimized over the regularization constant. For the level-0 component networks, the new "blob-wise Approximate Feature-Discovery" objective function was used (see below). We are still in the process of training the level-1 integration network.

The Feature-Discovery Objective Functions

In our previous work, we have used the objective function with a term $\varepsilon_P(\mathbf{w}) = \min_{x \in P}(-\log(y(x,\mathbf{w})))$ for each positive blob P, and $-\log(1-y(x,\mathbf{w}))$ at each negative position x. y is the network output and $\mathbf{w}$ is the parameter vector of the network. We have noted that this is an ad hoc choice, with no firm theoretical basis, and that some problems with the results of training may be due to the lack of rational basis for this objective.

In the past quarter we have invented objective functions that we call *Feature-Discovery* objectives. These measure the probability p_{old} that, if a pixel is detected by a network, the pixel will be within an example of the class of objects we wish to detect. This probability is estimated using Bayesian arguments. The exact expression given a particular prior for p_{old} is

$$\bar{p}_{old} = \int_0^1 (1-u)\left(2\frac{n_o}{N} + \sum_{x \in X_{\text{Pos}}} \frac{(1-u)y(x)}{1-uy(x)}\right)\prod_{\text{All } x}(1-uy(x))\,du \qquad \text{(EQ 1)}$$

where n_o is the number of pixels within the positive examples, N is the total number of pixels, and X_{Pos} is the set of all pixel locations within positive examples. Note that this is the mean value of p_{old}, since the training data does not determine an exact value, only a posterior distribution for it.

We have used the negative logarithm of Equation 1 for training on a toy problem. The integral was evaluated numerically. This is slow, but feasible.

Because Equation 1 is rather complex and its use is expensive, we derived an approximation to it. This is

$$\bar{p}_{old} \approx \frac{2n_o / N + \sum_{x \in X_{Pos}} y(x)}{2 + \sum_{\text{All } x} y(x)} \qquad \text{(EQ 2)}$$

As an objective function we have been using one minus this expression.

Note the intuitive interpretation of Equation 2: The sum in the numerator is the mean number of detections in the positive examples, while the sum in the denominator is the mean number of detections anywhere. The ratio of these would be a typical estimate of p_{old}, i.e., what fraction of the detected pixels were in positive examples, except the detections are not definite, so we use mean values. The simple ratio fails when there are no detections. In this case, Equation 2 becomes n_o / N, which is just the probability that a pixel chosen at random from the training data would be in a positive example. This is a reasonable choice if we have never seen an example of a detected pixel, since a detection is irrelevant information in that case. The 2's in the numerator and denominator arise naturally, but probably are not terribly critical. See the attached report, "Learning to Detect Characteristic Sub-Patterns," for details of the derivations, and for a description of an artificial problem on which this was tried. (This report has not been published or submitted for publication.) In the artificial problem, the objective successfully trained networks to detect sub-features which were not always present in the positive regions, even though the training procedure treats all positive objects the same, whether or not they have a feature.

Equations 1 and 2 are *pixel-wise* functions, which seems like a rational approach, since in application the network output will be evaluated on each pixel, and if it is large we would conclude that an object is likely to contain that pixel. However, these objectives can reward a network for multiple detections within a single object. This is not a bad thing, but it would be better if the network was rewarded for detecting fewer features in more objects, rather than more features in fewer objects. To accomplish this, it is easy to modify Equation 2 by replacing the probabilities of detecting pixels with the probabilities of detecting blobs. The positive blobs are determined in the training data. For negative blobs, we either choose to cut the negative portions of the image into squares of about the same size as the objects, or they are given to us by lower-resolution networks in a coarse-to-fine search system.

With the usual interpretation of the network's output y as the probability of the presence of some object or feature, the probability of detecting a blob i is $z_i = 1 - \prod_{x \in \text{Blob } i} (1 - y(x))$. This is simply one minus the probability of detecting no pixels in the blob. The resulting error function is

$$\bar{p}_{pbld} \approx \frac{2n_{pb} / N_b + \sum_{\text{Positive Blobs } i} z_i}{2 + \sum_{\text{All Blobs } i} z_i} \qquad \text{(EQ 3)}$$

where n_{pb} is the number of positive blobs and N_b is the total number of blobs. This is the mean probability that a given blob contains a positive, given that the feature detector detected at least one pixel in it. On the artificial problem this worked much better than Equation 2. For discovering the features in the second pattern tree artificial problem (described above), this tended to get stuck in very flat regions of weight space in which all blobs have high probability of being detected. To avoid this, we first trained for a small number of iterations with Equation 2, and then switched to Equation 3. This seemed to give good results almost all of the time.

Training with Errors in the Desired Outputs

In a previous ARPA project, objective functions were developed for classification tasks when there are errors in the desired outputs in the training data. In this case, the training data is a set of feature vectors, each associated with a desired output. The desired output is either a binary or one-of-N value. This is very different from the uncertain-object-position objective, since in that case we know that an object is present, but our knowledge of its position is poor. Here, there is no position, just a certain probability that the correct class for a feature vector should have been one thing, given the class in the so-called desired outputs. We spent a small amount of time refining this work, and working on a report about it. See the attached report, "Dealing with Errors in Training Data for Classification Problems." This is not directly relevant to our current artificial problems, but it may become relevant if we have problems in which the training data incorrectly classifies locations in the images, and doesn't merely give wrong positions for correctly classified objects. (This report has not been published or submitted for publication, since there is still a lot that can be developed along these lines.)

Multi-resolution Shape Features

Our previous work has focused on fairly low level features as input to our hierarchical neural network. For example, we have shown how one can apply Laplacian and oriented band-pass operators to construct a feature vector which is useful for detection of small objects (i.e. objects occupying only a few pixels in the image, such as buildings and microcalcifications). Important to note is that for these small objects, features describing details of shape are not particularly important for detection. However, when objects become larger, shape begins to play a more important role. For instance, we applied our hierarchical neural network, using low level features to the problem of detecting aircraft in aerial images. These aircraft are significantly larger than the buildings or microcalcifications we had considered previously (aircraft ranged from 20x20 to 50x50 pixels in size). Results indicate that low level features are not the most effective for detecting these larger objects having parts

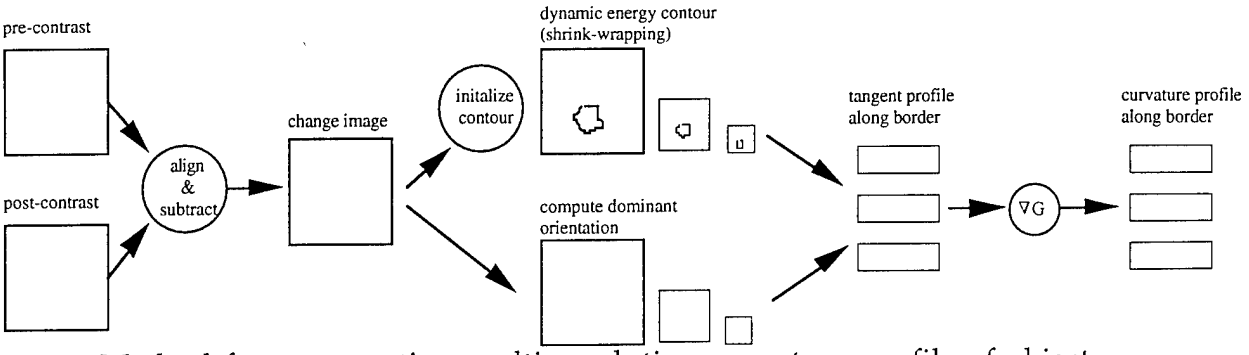

Figure 1: Method for constructing multi-resolution curvature profile of object border.

One approach for dealing with large objects is to automatically learn a set of features for the object class (see section on Feature Discovery). A second avenue we have been investigating is the use of higher level features which are more descriptive, in terms of object shape. For example, in breast imaging, high resolution MRI is often used diagnostically. Radiologists have found that the shape of the border is highly correlated with the probability and degree of malignancy of a lesion. Lesions which have smooth borders tend to be benign while lesions with rough, spiculated borders, are more likely malignant. One method of representing border shape is the curvature profile of the border at multiple scales. A distinction between benign and malignant lesions might therefore be represented in the multi-scale curvature profiles along a lesion border.

We have developed an algorithm for constructing a multi-scale curvature profile of a lesion (see figure 1). The first step in the algorithm is to construct a change image by subtracting pre and post contrast MR images. A contour is then placed around a suspect lesion and, using a dynamic contour algorithm, shrink-wraps around the lesion. The shrink wrapped contour defines a path over which a tangent profile can be constructed using the output of multi-scale orientation operators. Using the first derivative of a Gaussian, the tangent profiles are transformed into curvature profiles. Figure 2 shows results for two different lesions. The border of lesion 1 is smoother than that of lesion 2 and therefore lesion 2 is more likely cancerous. This can be seen in the curvature profiles and also the variance of the profiles (see table 1). The next planned step is to extract features from these multi-scale curvature profiles (these feature could be the variances, Fourier Descriptors, or even the signal itself) as an input into our hierarchical neural network classifier . The network would then integrate low resolution shape descriptors with higher resolution shape descriptors for object detection and classification. We then plan to extend the system in order to consider the ATR example of detecting aircraft in aerial images.

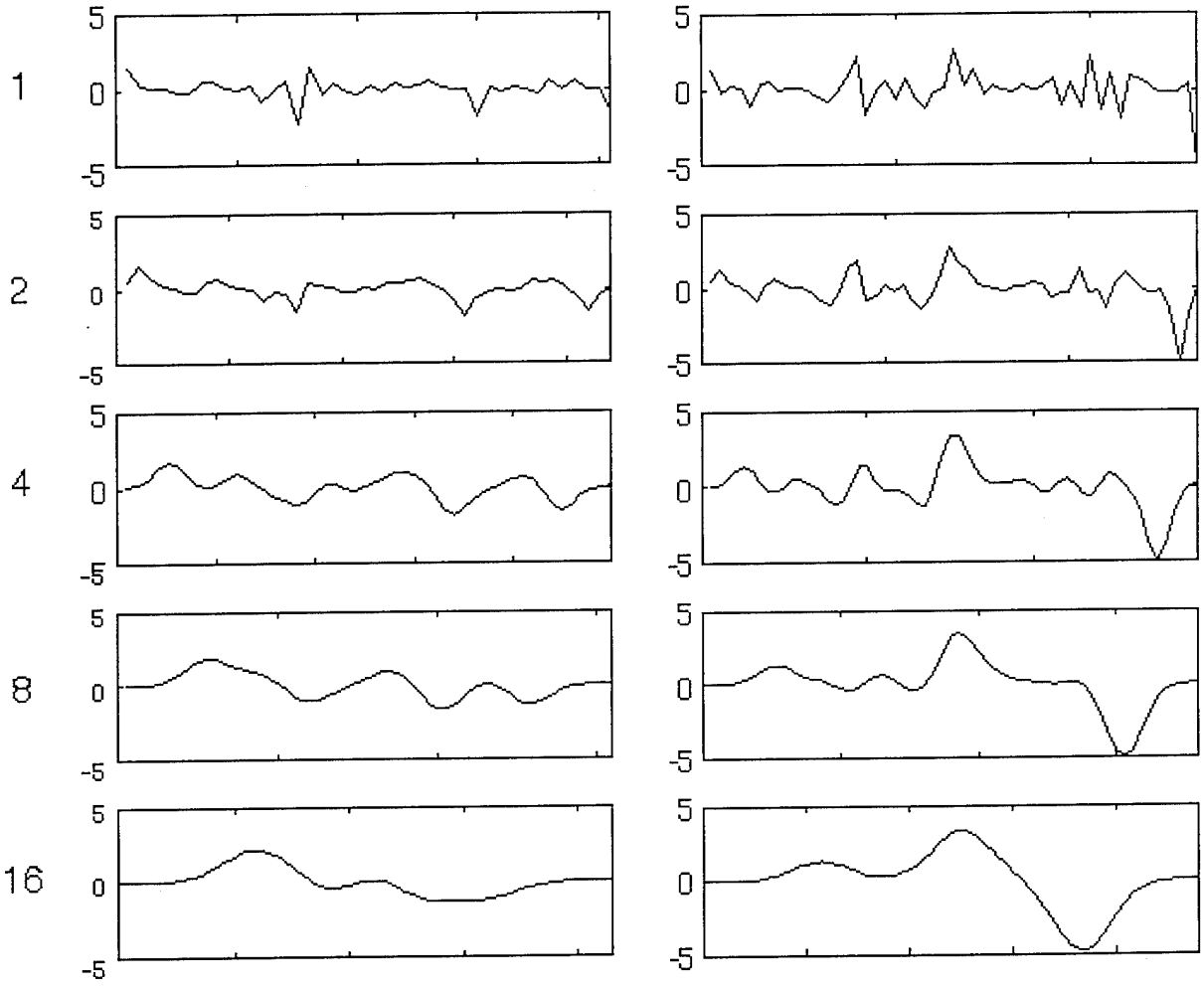

Figure 2: Curvature profiles for two different breast (MRI) lesions

Table 1: Variance of curvature profiles for lesions 1 & 2

scale	lesion 1	lesion 2
1	0.45	1.34
2	0.40	1.24
4	0.66	1.91
8	0.79	2.63

16	0.86	3.68

Mercury Computers

As mentioned in our last report, we have initiated talks with Mercury Computers regarding possible commercial applications of our Neural Network/Pyramid Software to problems in ATR and biomedical imaging. Mercury is a manufacturer of high-end parallel computer hardware, supplying platforms for both ATR and biomedical image processing. Since our last report, Mercury has visited Sarnoff (Mercury visitors included Bruce Beck, VP, Network Systems; Don Berry, Director, Commercial Products Group; Chris Stakutis, Product Manager) and is interested in our technology. They requested a second meeting in June/July to discuss the technologies and a potential partnership in further detail.

Acceptance of paper in Neural Networks, Special Issue on ATR

Our paper entitled "Integrating Neural Networks with Image Pyramids to Learn Target Context" was accepted for publication in the Special ATR issue of *Neural Networks*

Acceptance of paper for International Conference on Image Processing (ICIP95)

Our paper entitled "A Hierarchical Neural Network Architecture that Learns Target Context: Applications to Digital Mammography" was accepted for publication and presentation at ICIP95.

NIPS Paper submitted

A paper titled "Training neural networks with uncertain object positions" was submitted for presentation at the 1995 Neural Information Processing Systems Conference.

Summary of Substantive Information Derived from Special Events:

None.

Problems Encountered and/or Anticipated:

Because of the advances we are making, we have spent ahead of schedule, and expect to have spent the currently provided funds by the middle of June. Work will stop at that time, unless we decide to spend in advance.

Action Required by the Government:

We have contacted COTR Tom McKenna, asking for a letter stating the probability of our receiving the next funding increment. We will spend in advance only if this probability is quite high.

ARPA IU program management is expected to visit us on July 7. We are preparing a presentation of our progress and the project status.

Financial Status

1. Amount currently provided on contract: $425,740
2. Expenditures and commitments to date: $412,682
3. Funds required to complete work: $251,130